

Enhanced AMBTC for Image Compression using Block Classification and Interpolation

S. Vimala
Dept. of Comp. Science
Mother Teresa Women's
University
Kodaikanal – 624 102

M. Sathya
Dept. of Comp. Sci.
Mother Teresa Women's
University
Kodaikanal – 624 102

K. Kowsalya Devi
Dept. of Comp. Sci.
Mother Teresa Women's
University
Kodaikanal – 624 102

ABSTRACT

Block Truncation Coding (BTC) is a fast moment preserving lossy image compression technique. This paper proposes a two-level compression scheme. In the first level of compression, the blocks are categorized into high-detail blocks and low-detail blocks. The high detail blocks are stored or transmitted in the form of a bit plane (16 bits), and two statistical moments; high mean and low mean (16 bits). Only the mean values are stored for low detailed blocks. In second level of compression, the interpolation technique is incorporated as part of the proposed method for further bit rate reduction. The performance of the method is tested with standard images Lena, Cameraman Boats, Bridge, Baboon and KUSH. The proposed scheme achieves better coding efficiency with a slight degradation in PSNR when compared to existing methods.

General Terms

Image Compression

Keywords

image compression; high-detail block; low-detail block; interpolation; bpp; PSNR;

1. INTRODUCTION

Generally images occupy more storage space and take more time to get transmitted across network. Image compression technique is used to reduce the cost of storing, retrieving and transmission of images across a network [1]. One of the fundamental goals of image compression is to reduce the size of data needed to represent the image while maintaining the necessary information [2]. There are two types of Image compression techniques: lossless techniques and lossy techniques. A lossy compression technique provides much higher compression ratio than lossless scheme.

The performance of any compression technique is measured in terms of 1.Compression ratio 2.Signal-to-Noise Ratio (quality of the reconstructed image) and 3.Encoding and Decoding Speed [3].

In lossy image compression method, the redundant information can be compressed [4]. One such lossy image compression technique is Block Truncation coding (BTC) technique. BTC is one of the recent techniques used to compress image data. The computational complexity is less in terms of BTC when compared to other techniques. A

compression rate of 2 bpp is achieved with BTC [5] & [6]. The compressed image is only an approximation of the original image. Though some data is removed, for Human Visual System, there is no much difference between the original and the compressed image. In encoding process, for each input image block of size 4x4 pixels, a bit plane is formed by replacing each pixel value with either 1 or 0. When the pixel value is greater than mean, it is coded as 1, otherwise 0. Two statistical moments of size 16 bits are preserved along with the bit plane [7].

The bit-rate achieved with BTC can further be reduced by reducing the amount of data needed to represent the Bitplane by dropping few bits. The dropped bits can be generated at the decoding stage by using the Interpolation technique [8]. By using interpolation method, the size of bitplane is reduced from 16 bits to just 8 bits leading to a bitrate of 1.75 bpp reduced from 2 bpp. The total number of bits is 24 for each block and the bit rate is reduced from 2 to 1.5 bits/pixel. In BTC, the statistical overheads: the standard Mean and standard deviation are preserved for each input block. The truncated block of the BTC is the one-bit output of the quantizer for every pixel in the block. A simple variant of BTC called Absolute Moment Block Truncation Coding (AMBTC) was presented by Lema and Mitchell, and it preserves the higher mean and lower mean of the blocks [9]. The interpolative techniques are used to code the compressed bit plane of BTC [10, 11]. An algorithm for preserving moments which results in less mean square error (MSE) was proposed in [12].

To reduce the bitrate further, few adaptive Block Truncation Coding techniques have been evolved: A four level four level quantizer has been designed in [13] and an idea of Multilevel Quantizer has been proposed in [14].

The difference between the original image and the reconstructed image is called Mean Square Error (MSE) and is calculated using the Eq. (1). The Quality of the reconstructed image is measured using Peak Signal to Noise Ratio (PSNR) and is computed using the Eq. (2). PSNR is the inverse of MSE.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - x_i)^2 \quad (1)$$

Where y_i is the reconstructed pixel value, x_i is the original pixel value and N is the number of Pixels in an image.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (2)$$

The performance of an image compression technique is measured in terms of the PSNR and bpp. The PSNR must be high and bpp must be low.

1.1. Standard BTC

In BTC Algorithm, the image is divided in to non-overlapping blocks of size 4 x 4 pixels. The mean \bar{x} is computed using Eq. (3) for each block.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (3)$$

Bit-plane Generation: Each pixel value in the block is compared against \bar{x} . If the value is greater than or equal to the mean value, it is coded as 1 otherwise 0.

The standard deviation σ is computed using the Eq. (4) to generate the statistical moments.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (4)$$

Where x_i represents the i^{th} pixel value of the image block and n is the total number of pixels in that block [13].

The two statistical moments H and L are computed using the Eq. (5) and Eq. (6).

$$H = \bar{x} + \sigma \sqrt{\frac{p}{q}} \quad (5)$$

$$L = \bar{x} - \sigma \sqrt{\frac{q}{p}} \quad (6)$$

Where p and q are the number of 0's and 1's in the compressed bit plane respectively. In the decoder, an image block is reconstructed by replacing 1s in the bit plane with H and the 0s with L .

1.2. Absolute Moment Block Truncation Coding (AMBTC)

Lema and Mitchell presented a simple and fast variant of BTC named Absolute Moment Block Truncation Coding (AMBTC). In this method, the higher mean \bar{x}_h and lower mean \bar{x}_l are preserved instead of the mean and standard deviation values. Pixels in an image block are then classified into two groups of values. One group (higher range) comprising of gray levels which are greater than or equal to the mean (\bar{x}) and the remaining gray levels are brought into another group (lower range). The mean \bar{x}_h of higher range and \bar{x}_l of the lower range are calculated using the Eq. (7) and Eq. (8).

$$\bar{x}_l = \frac{1}{(m-q)} \sum_{x_i < \bar{x}} x_i \quad (7)$$

$$\bar{x}_h = \frac{1}{(q)} \sum_{x_i \geq \bar{x}} x_i \quad (8)$$

where q is the number of pixels whose gray levels are greater than or equal to \bar{x} . Then a two level quantization is performed for all the pixels in that block to form a bit plane of 1's and 0's. The encoder generates \bar{x}_l , \bar{x}_h and bit plane for each block, thus leading to 2 bpp like BTC. AMBTC involves less number of computations, when compared to BTC as standard deviation involves more multiplications. AMBTC yields high quality of reconstructed images.

1.3. Interpolative Technique

The boldfaced components as in Fig.1 are dropped and the size of the bit plane is 8x1=8 bits rather than 16 bits [14].

X_1	X_2	X_3	X_4
X_5	X_6	X_7	X_8
X_9	X_{10}	X_{11}	X_{12}
X_{13}	X_{14}	X_{15}	X_{16}

Fig.1 The pattern of dropping bits

The existing 8bits are transformed into gray levels based on the quantizers in decoding phase. The dropped bits are recovered by taking the arithmetic mean of the adjacent pixel values as given in Eq. (9).

$$\left. \begin{aligned} X_i &= \frac{1}{3}(x_{i-1} + x_{i+1} + x_{i+4}) \text{ for } i = 2 \\ X_i &= \frac{1}{2}(x_{i-1} + x_{i+4}) \text{ for } i = 4 \\ X_i &= \frac{1}{3}(x_{i-4} + x_{i+1} + x_{i+4}) \text{ for } i = 5 \\ X_i &= \frac{1}{4}(x_{i-4} + x_{i-1} + x_{i+1} + x_{i+4}) \text{ for } i = 7,10 \\ X_i &= \frac{1}{3}(x_{i-4} + x_{i+1} + x_{i+4}) \text{ for } i = 12 \\ X_i &= \frac{1}{2}(x_{i-4} + x_{i-1}) \text{ for } i = 13 \\ X_i &= \frac{1}{3}(x_{i-5} + x_{i+4} + x_{i+1}) \text{ for } i = 15 \end{aligned} \right\} \quad (9)$$

2. PROPOSED TECHNIQUE

The proposed method is based on AMBTC technique for generating the bit plane and the statistical moments, high-mean and low-mean. The compression is done in two levels. In first level of compression, block classification is done. In second level of compression, the interpolation method is used.

At the first level of compression, each input block is categorized into either a high detail block or a low detail block. To categorize, the mean of each block is computed using Eq. (3). The sum of differences between the mean and the individual pixel value of each block is computed using Eq. (10).

$$Diff = \sum_{i=1}^m abs(x_i - \bar{x}) \quad (10)$$

If the difference is greater than a threshold value, the block is categorized as high detail block otherwise low detail block. In low detail block, the pixel values are closer to each other. Hence for a low detail block, only the mean value is preserved. But for a high detail block, a bit plane (16 bits) and two statistical moments: hmean and lmean are stored as the difference between the pixel values is high. Therefore the number of bits required to store a high detail block is high and is low for a low detail block.

Encoding Algorithm

Level 1

Step1: Input the image of size m x m pixels.

Step2: Split the image into blocks of size 4 x4 pixels.

Step3: For each block, perform the following.

- Compute the mean using “Eq. (3)”.
- Compute the sum of differences using “Eq. (10)”.
- if $Diff \geq \text{Threshold}$, declare it as a high detail block
else

it is a low detail block

(d) if high detail block

i. generate the bit plane

ii. compute \bar{x}_h using “Eq. (7)” and \bar{x}_l using “Eq. (8)”.

iii. store/transmit the set {bit plane, \bar{x}_h and \bar{x}_l }.

(e) if low detail block, store and transmit only the mean.

At the second level of compression, the interpolation method is incorporated as part of the proposed method. Certain elements of bit plane as shown in fig.1 are dropped leading to bit rate reduction.

Level 2

Step1: For each high detail block, perform the following

(a) Drop the elements of the bit plane as in Fig. 1.

The compressed image is either stored or transmitted in the form of compressed bit plane (8 bits), high mean (8 bits) and low mean (8 bits) for high detail blocks and, only the block-mean values are stored for the low detail block. The compressed image is reconstructed (decoded) at the receiving end as follows: For high detail blocks, the 1's and 0's in the

bit plane are replaced with \bar{x}_h and \bar{x}_l respectively. Then the dropped elements are regenerated using the Eq. (11). The low detail blocks are reconstructed only with the mean values.

Decoding Algorithm

Step1: Reconstruct the image as follows:

Step2: if high detail block,

(a) replace all 1's in the bit plane with \bar{x}_h and 0's with \bar{x}_l .

(b) interpolate the dropped elements using the “Eq. (11)”.

Step3: if low detail block, generate a block of 4x 4 pixels with mean for all pixel values.

$$\left. \begin{aligned} X_i &= \frac{1}{2}(x_{i-1} + x_{i+4}) \text{ for } i = 2,5 \\ X_i &= \frac{1}{2}(x_{i+1} + x_{i+4}) \text{ for } i = 3,8 \\ X_i &= \frac{1}{2}(x_{i+1} + x_{i+4}) \text{ for } i = 12,15 \\ X_i &= \frac{1}{2}(x_{i+1} + x_{i+4}) \text{ for } i = 9,14 \end{aligned} \right\} \quad (11)$$

By categorizing the blocks into high detail block or low detail block, the coding efficiency can be improved. In BTC /

AMBTC, each block requires 32 bits (16 for bit plane, 16 for two statistical moments). The bpp achieved is $32 / 16 = 2$ bpp, where a block contains 16 elements. In the proposed method, the high detail blocks are stored in the form of bit plane and two statistical moments. Only the mean value is stored for the low detail block. The bpp achieved for a low detail block is $8/16$, which is just 0.5. The bpp varies for each image depending on the number of high and low detail blocks. If $n1$ is the number of high detail blocks and $n2$ is the number of low detail blocks, the average bpp is computed using the Eq. (12).

$$\text{bpp} = ((n1 * 32) + (n2 * 16)) / (256 * 256) \quad (12)$$

where $256 * 256$ is the number of pixels in an image. The bit rate is further reduced when interpolation method is used.

For the cameraman image,

Number of high detail blocks, $n1 = 587$

Number of low detail blocks, $n2 = 3509$

The average bpp is computed as follows,

The total number of bits after level1 (block classification):

$$\begin{aligned} n &= (587 * 32) + (3509 * 8) = 46856 \\ \text{bpp} &= 46856 / (256 * 256) = 0.72 \text{ bpp.} \end{aligned}$$

The total number of bits after level 2 (interpolation):

$$\begin{aligned} N1 &= (587 * 24) + (3509 * 8) = 42160 \\ \text{bpp} &= 42160 / (256 * 256) = 0.64 \text{ bpp.} \end{aligned}$$

Similarly, the bpp is calculated for different images with different distribution of gray shades. The average bpp for the entire image depends on the number of low and high detail blocks and is different for each image.

3. RESULTS AND DISCUSSION

The proposed method is tested with Lena, Cameraman, Boats, Bridge, Baboon and Kush images. The input images taken for the study are given in Fig. 2. The results obtained in terms of PSNR and bpp with the proposed method are compared with that of the BTC and AMBTC methods in Table.1. The results are generated for three different threshold values.



(a) Lena



(b) Cameraman



(c) Boats



(d) Bridge



(e) Baboon



(f) Kush

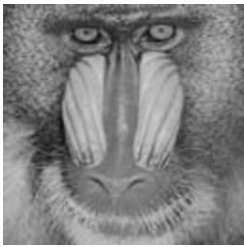
Fig.2. Input images taken for the study

The results in Table.1 show that there is a significant improvement in the coding efficiency, i.e. from 2 to 0.7, 0.5 and 0.48 for Lena image at the cost of PSNR. A minimum of 0.70 bpp is achieved for Lena image with the threshold value of 50. It is observed that the reduction in bpp is due to less number of high detail blocks in any image. Number of high detail blocks is high in Boats image and the bpp obtained is 1.12, which is a maximum value. The bpp is reduced further

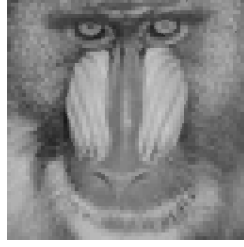
when the threshold values are raised from 50 to 100 and 150. The number of high detail blocks will be less when the threshold values are increased. The bpp will also be decreased if the number of high detail blocks decreases. A minimum of 0.40 bpp is achieved for Baboon image with the threshold value of 150, which is a significant improvement. The reconstructed images are given in Fig.3.

Table.1 The PSNR and the bpp values obtained with respect to BTC, AMBTC and the proposed method for different Threshold Values.

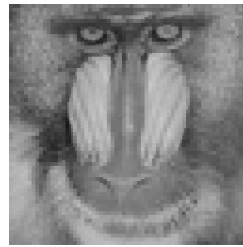
Images	BTC		AMBTC		Threshold 50		Threshold 100		Threshold 150	
	PSNR	bpp	PSNR	bpp	PSNR	Bpp	PSNR	bpp	PSNR	bpp
Lena	34.62	2.00	34.85	2.00	32.27	0.70	31.64	0.55	30.91	0.48
Cameraman	32.07	2.00	32.13	2.00	28.73	0.70	28.53	0.57	28.24	0.50
Boats	32.77	2.00	33.15	2.00	28.25	1.12	27.85	0.82	27.15	0.60
Bridge	30.68	2.00	30.94	2.00	29.80	0.81	29.52	0.64	29.10	0.56
Baboon	37.14	2.00	37.46	2.00	35.31	0.78	33.25	0.47	32.26	0.40
Kush	35.41	2.00	35.55	2.00	33.31	0.73	32.30	0.52	31.44	0.45



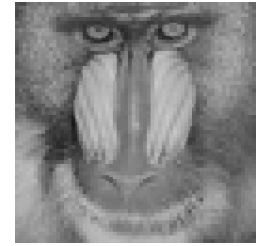
Original
(bpp : 8.00)



Threshold 50
(bpp: 0.78)



Threshold 100
(bpp : 0.47)



Threshold 150
(bpp: 0.78)



Original
(bpp : 8.00)



Threshold 50
(bpp : 0.73)



Threshold 100
(bpp : 0.52)



Threshold 150
(bpp : 0.45)

Fig.3 Reconstructed images obtained with different Threshold values 50,100 and 150.

4. CONCLUSION

A two-stage compression is achieved with the proposed method. In the first stage, the bit rate is reduced by categorizing the blocks into high detail and low detail blocks. The interpolation technique is used in the second stage to reduce the bit-rate further. 75% of compression is achieved with the existing methods BTC and AMBTC. A compression rate of 95% is achieved with the proposed method, which is a significant improvement. The quality of reconstructed images depends on the selection of threshold values. This technique is best suitable for hand-held devices and may also be extended to color images.

5. REFERENCES

- [1] K.Somasundram and P.Sumitra, "RGB & Gray Scale Component on MPQ-BTC in image Compression", *International Journal on Computer Science and Engineering*, Vol. 3, No. 4, April 2011.
- [2] K.Somasundaram and S.Vimala, "Codebook Generation for vector Quantization with Edge features", *CiiT, International Journal of Digital Image Processing*, 2010.
- [3] Sonal and DineshKumar, "A Study of various image compression technique", *National Conference on Challenges and Opportunities in Information Technology*, Hisar, 2007.
- [4] Pham, D.Premkumar, A. Madaikumar, "Error Detection and Correction in Communication Channels Using Inverse Gray RSNS Codes", *IEEE Transaction on Communications*.
- [5] S.Vimala, P.Uma, B.Abidha, "Improved Adaptive Block Truncation Coding for Image Compression", *International Journal of Computer Applications*, Vol.19-No.7, April 2011.
- [6] Aditya Kumar and Pradeep Singh, "Enhanced Block Truncation coding for Gray Scale Image", *International Journal of Computer Technology and Application*, Vol 2 (3), 525-529.
- [7] Doaa Mohammed, Fatma Abou-Chadi, "Image compression using Block Truncation Coding", *Multidisciplinary journals in science and Technology*, JSAT, February 2011.
- [8] Elghanai.M. Rhoma, Abdosllam.M. Abobaker, "Mean Square Error Minimization Using Interpolative Block Truncation Coding Algorithms", *Proceedings of the 2nd International Conference on Education Technology and Computer*, Vol.5, 2010.
- [9] Lema, M.D. and O.R. Mitchell, "Absolute Moment Block Truncation Coding and its Application to Color Images", *IEEE Trans. On Communications*, pp: 1148-1157, 1984.
- [10] K.Somasundaram and I.Kaspar Raj, "A New Predictive and Interpolative Image Compression Scheme Based on Block Truncation Coding", *International Journal of Soft Computing*, Vol. 2, No. 1, 2007.
- [11] Hengfu Yang and Jianping Yin, "An Adaptive Block Truncation Coding Algorithm Based on Data Hiding", *Information Technology Journal* 11 (5): 647-650, 2012.
- [12] Halverson D.R., "A generalized BTC Algorithm for Image Compression", *IEEE Transactions on Acoustics, Speech, Signal Processing*, Vol. ASSP-32, pp. 664-668, 1984.
- [13] Lucas Hui, "An Adaptive Block Truncation Coding for Image Compression", *IEEE Transactions*, CH2847-2, pp. 2233-2236, 1990.
- [14] Madhu Shandilya and Rajesh Shandilya, "Implementation of Absolute Moment Block Truncation Coding Scheme Based on Mean Square Error Criterion", *Proceedings of the SDR 03 Technical Conference and Product Exposition*, 2003.